Polynomial spline interpolation problem

Virgilius-Aurelian Minuță

Babeș-Bolyai University of Cluj-Napoca
Faculty of Mathematics and Computer Science
Department of Mathematics

Cluj-Napoca

3rd February 2021

Introduction

The polynomial interpolation problem

Let $f : I \to \mathbb{R}$ be a function, where $I = [a, b] \subset \mathbb{R}$.
Let Δ be a subdivision upon the interval $[a, b]$

$$\Delta : a = x_1 < x_2 < \ldots < x_{n-1} < x_n = b$$

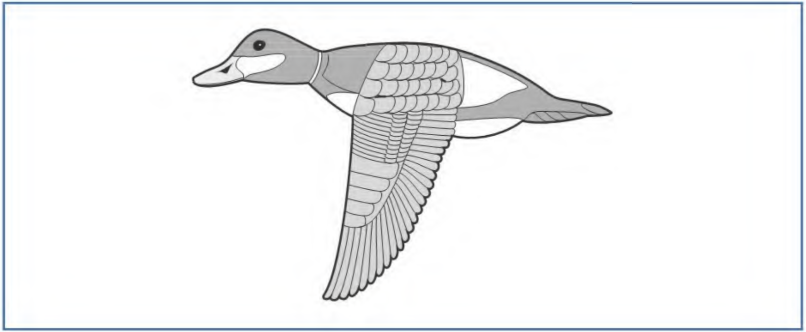
Given the values of f, $f_i = f(x_i)$, $i = \overline{1, n}$, find a function φ in a class of "approximations" Φ such that

$$\varphi(x_i) = f_i, \quad i = \overline{1, n}.$$

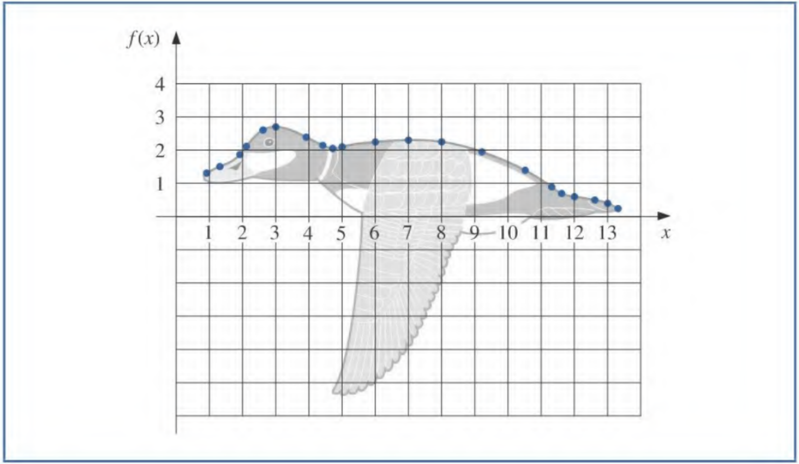
Example

If $\Phi = \mathbb{P}_m$ - the set of polynomials of degree at most m; we deal with polynomial interpolation. The interpolation problem is called here Lagrange interpolation and Hermite interpolation, respectively.

Application - Lagrange polynomial interpolation

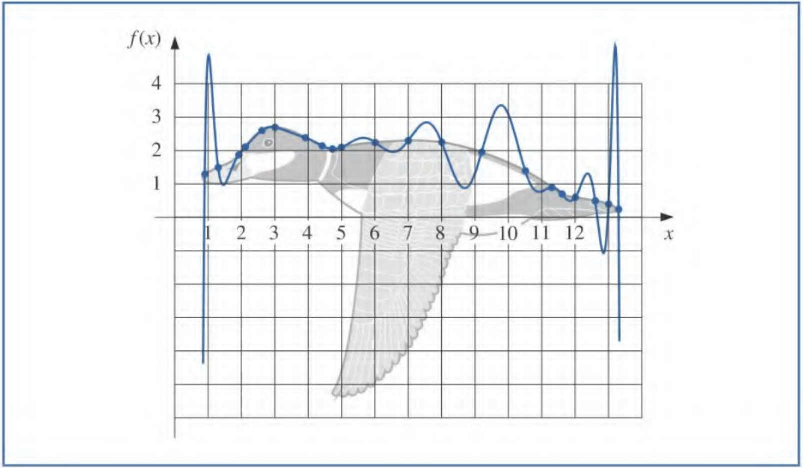


Application - Lagrange polynomial interpolation



x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

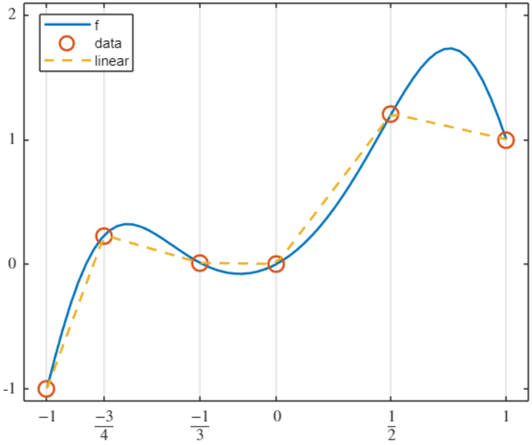
Application - Lagrange polynomial interpolation



Polynomial spline interpolation

Idea

- use low-degree polynomials on each subinterval $[x_i, x_{i+1}]$, $i = \overline{1, n-1}$



Polynomial spline interpolation

Polynomial spline functions

- Let

$$\Phi = \mathbb{S}_m^k(\Delta) := \{s \mid s \in C^k([a,b]),\ s|_{[x_i, x_{i+1}]} \in \mathbb{P}_m,\ i = \overline{1, n-1}\},$$

  be the space of spline functions of degree $m \geq 0$ and smoothness class k relative to the subdivision Δ, where $k \in \mathbb{N} \cup \{-1\}$.
- We set $k = -1$ if we allow discontinuities at the joints.
- If $k = m$, then the functions $s \in \mathbb{S}_m^k(\Delta)$ are polynomials.

Linear spline interpolation

Linear spline interpolation

- For $m = 1$ and $k = 0$ we obtain linear splines.
- We want $s \in \mathbb{S}_1^0(\Delta)$ such that $s(x_i) = f_i$, $i = \overline{1, n}$.
- Solution: On the interval $[x_i, x_{i+1}]$

$$s(x) = f_i + (x - x_i) f[x_i, x_{i+1}]$$

- Interpolation error on the interval $[x_i, x_{i+1}]$:

$$|f(x) - s(x)| \leq \frac{(\Delta x_i)^2}{8} \max_{x \in [x_i, x_{i+1}]} |f''(x)|$$

- Interpolation error:

$$||f(x) - s(x)||_\infty \leq \frac{1}{8} |\Delta|^2 ||f''||_\infty$$

MATLAB Command

```
y=interp1(xi,fi,x,'linear')
```

Linear spline interpolation

Exercise 1

Write a MATLAB script that plots the graph of $f(x) = x + sin \pi x^2$ and computes the linear spline interpolant using the `interp1` command.

Solution

Linear spline interpolation

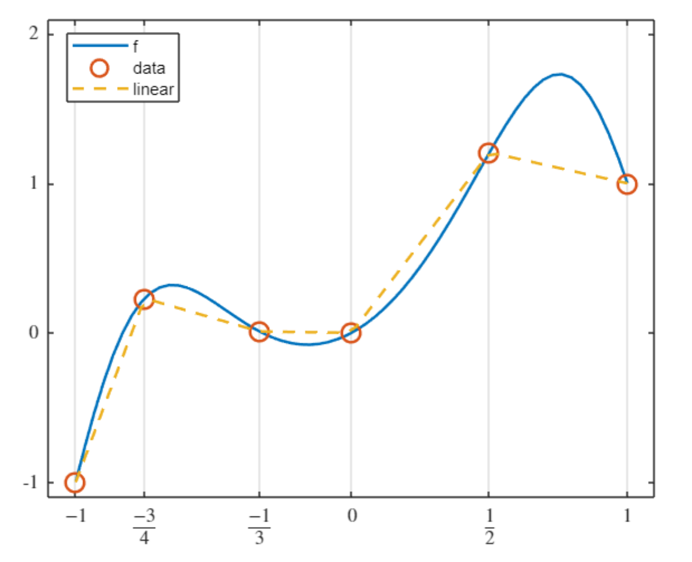
Solution

```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=[-1,-3/4,-1/3,0,1/2,1];
fi=f(xi);
x=linspace(-1,1,60);
y_f=f(x);
y_lin=interp1(xi,fi,x,'linear');
plot(x,y_f,'-',xi,fi,'o','MarkerSize',10);
hold on
plot(x,y_lin,'--')
xticks(xi)
yticks([-1,0,1,2]);
xticklabels({'$-1$','$\frac{-3}{4}$','$\frac{-1}{3}$','$0$','$\frac{1}{2}$','$1$'})
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-1.1, 1.1, -1.1, 2.1])
legend('f','data','linear','Location','best')
hold off

function result = f(x)
result = x+sin(pi*x.^2);
end
```

Linear spline interpolation

Solution



Interpolation by cubic splines

Interpolation by cubic splines when $s \in \mathbb{S}_3^1(\Delta)$

- Continuity of the first derivative can be enforced by prescribing values for the first derivative at each point x_i.
- Let $m_1, m_2, \ldots, m_n$ be arbitrary given numbers.
- Denote $s(x)|_{[x_i, x_{i+1}]} = p_i(x), \quad i = \overline{1, n-1}$.
- We select p_i to be the (unique) solution of a Hermite interpolation problem:

$$\begin{array}{lll} p_i(x_i) = f_i & p_i(x_{i+1}) = f_{i+1} & i = \overline{1, n-1} \\ p_i'(x_i) = m_i & p_i'(x_{i+1}) = m_{i+1} & \end{array}$$

- Solution in Taylor form, for $x_i \leq x \leq x_{i+1}$:

$$p_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3$$

where

$$\begin{array}{llllll} c_{i,0} & = & f_i & c_{i,1} & = & m_i \\ c_{i,2} & = & \frac{f[x_i, x_{i+1}] - m_i}{\Delta x_i} - c_{i,3}\Delta x_i & c_{i,3} & = & \frac{m_{i+1} + m_i - 2f[x_i, x_{i+1}]}{(\Delta x_i)^2} \end{array}$$

Interpolation by cubic splines

Piecewise cubic Hermite interpolation

- Here one sets $m_i = f'(x_i)$ (assuming that these derivative values are known).
- Interpolation error the interval $[x_i, x_{i+1}]$:

$$|f(x) - p_i(x)| \leq \left(\frac{1}{2}\Delta x_i\right)^4 \max_{x\in[x_i,x_{i+1}]} \frac{|f^{(4)}(x)|}{4!}$$

- Interpolation error:

$$||f(x) - s(x)||_\infty \leq \frac{1}{384}|\Delta|^4||f^{(4)}||_\infty$$

MATLAB Command

`y=interp1(xi,fi,x,'pchip')` - shape-preserving piecewise cubic interpolation $s \in \mathbb{S}_3^1(\Delta)$

We can also directly call the command `y=pchip(xi,fi,x)`

Interpolation by cubic splines

Exercise 2

Update the script from Exercise 1, to also compute a piecewise cubic interpolant using the `interp1` command.

Solution

Interpolation by cubic splines

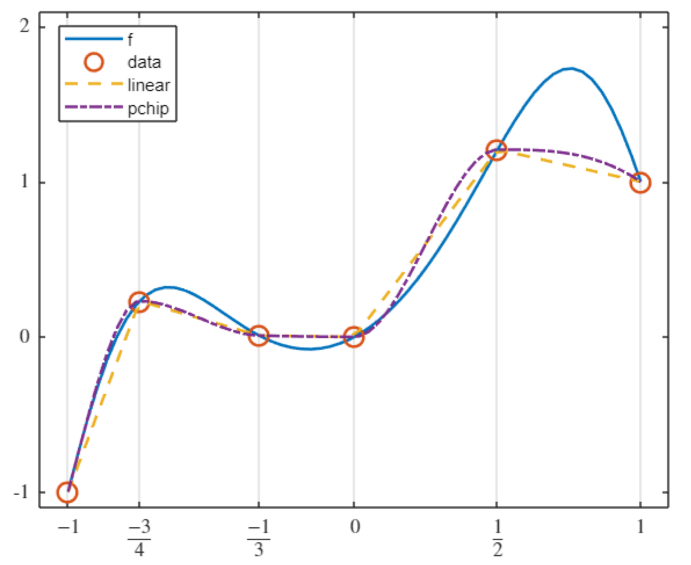
Solution

```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=[-1,-3/4,-1/3,0,1/2,1];
fi=f(xi);
x=linspace(-1,1,60);
y_f=f(x);
y_lin=interp1(xi,fi,x,'linear');
y_chip=interp1(xi,fi,x,'pchip');
plot(x,y_f,'-',xi,fi,'o','MarkerSize',10);
hold on
plot(x,y_lin,'--',x,y_chip,'-.')
xticks(xi)
yticks([-1,0,1,2]);
xticklabels({'$-1$','$\frac{-3}{4}$','$\frac{-1}{3}$','$0$','$\frac{1}{2}$','$1$'})
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-1.1, 1.1, -1.1, 2.1])
legend('f','data','linear','pchip','Location','best')
hold off
```

```
function result = f(x)
result = x+sin(pi*x.^2);
end
```

Interpolation by cubic splines

Solution



Interpolation by cubic splines

Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

- We have that

$$p''_{i-1}(x_i) = p''_i(x_i), \quad i \in \overline{2, n-1}$$

- Hence

$$2c_{i-1,2} + 6c_{i-1,3}\Delta x_{i-1} = 2c_{i,2}, \quad i \in \overline{2, n-1}$$

- We obtain the linear system (for $i \in \overline{2, n-1}$):

$$\Delta x_i m_{i-1} + 2(\Delta x_{i-1} + \Delta x_i)m_i + (\Delta x_{i-1})m_{i+1} = 3(\Delta x_i f[x_{i-1}, x_i] + \Delta x_{i-1} f[x_i, x_{i+1}])$$

- This is a system of $n-2$ linear equations with n unknowns $m_1, m2, \ldots, m_n$.
- By choosing m_1 and m_n, the system can be solved.

MATLAB Command

```
y=spline(xi,fi,x)
```

If `fi` has two more values than `xi` has entries, the first and last value play the role of m_1, m_n.

Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

Complete (clamped) spline

- We take $m_1 = f'(a)$ and $m_n = f'(b)$
- Interpolation error, if $f \in C^4([a,b])$:

$$||f^{(r)}(x) - s^{(r)}(x)||_\infty \le c_r|\Delta|^{4-r}||f^{(r)}||_\infty, \quad r = \overline{0,3},$$

where $c_0 = \frac{5}{384}$, $c_1 = \frac{1}{24}$, $c_3 = \frac{3}{8}$ and c_3 is a constant depending on the ration $\frac{|\Delta|}{\max_i \Delta x_i}$.

Complete (clamped) spline

Exercise 3

Write a script that plots the graph of the $sin(x)$ function and compute a piecewise Hermitian interpolation together with a complete spline interpolation using the commands `pchip` and `spline`.

Solution

Complete (clamped) spline

Solution

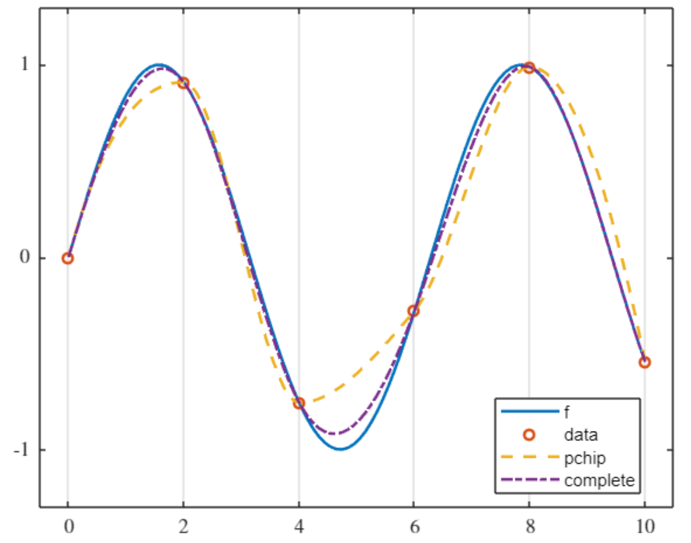
```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=0:2:10;
fi=f(xi);
x=linspace(0,10,100);
y_f=f(x);
y_pchip=pchip(xi,fi,x);
y_complete=spline(xi,[cos(0),fi,cos(10)],x);
plot(x,y_f,'-',xi,fi,'o','MarkerSize',5);
hold on
plot(x,y_pchip,'--',x,y_complete,'-.')
xticks(xi)
yticks([-1,0,1]);
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-0.5, 10.5, -1.3, 1.3])
legend('f','data','pchip','complete','Location','best')
hold off

```

```
function result = f(x)
result = sin(x);
end
```

Complete (clamped) spline

Solution



Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

Natural cubic spline

- We enforce $s''(a) = s''(b)$.
- We obtain 2 additional equations in the system

$$\begin{aligned} 2m_1 + m_2 &= 3f[x_1, x_2] \\ m_{n-1} + 2m_n &= 3f[x_{n-1}, x_n] \end{aligned}$$

- Advantage: No need to know derivative values.
- Disadvantage: Accuracy degradation at the endpoints (unless $f''(a) = f''(b) = 0$).

Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

"Not-a-knot spline" (Carl de Boor)

- We enforce $p_1 \equiv p_2$ and $p_{n-2} \equiv p_{n-1}$
- This means that the first and last "knots" x_2 and x_{n-1} are inactive
- We again obtain 2 additional equations in the system:

$$\begin{aligned} \Delta x_2 m_1 + (\Delta x_2 + \Delta x_1) m_2 &= \gamma_1 \\ (\Delta x_{n-1} + \Delta x_{n-2}) m_{n-1} + \Delta x_{n-2} m_n &= \gamma_2, \end{aligned}$$

where

$$\begin{aligned} \gamma_1 &= \frac{1}{\Delta x_2 + \Delta x_1}\left(f[x_1, x_2]\Delta x_2 (3\Delta x_1 + 2\Delta x_2) + (\Delta x_1)^2 f[x_2, x_3]\right) \\ \gamma_2 &= \frac{1}{\Delta x_{n-1} + \Delta x_{n-2}}\left((\Delta x_{n-1})^2 f[x_{n-2}, x_{n-1}] + (3\Delta x_{n-1} + 2\Delta x_{n-2})\Delta x_{n-2} f[x_{n-1}, x_n]\right). \end{aligned}$$

MATLAB Command

`y=interp1(xi,fi,x,'spline')` or `y=spline(xi,fi,x)`

“Not-a-knot spline” (Carl de Boor)

Exercise 4

Update the script of Exercise 3, such that it plots a complete spline interpolation and a deBoor interpolation using the commands `interp1` and `spline`.

Solution

“Not-a-knot spline” (Carl de Boor)

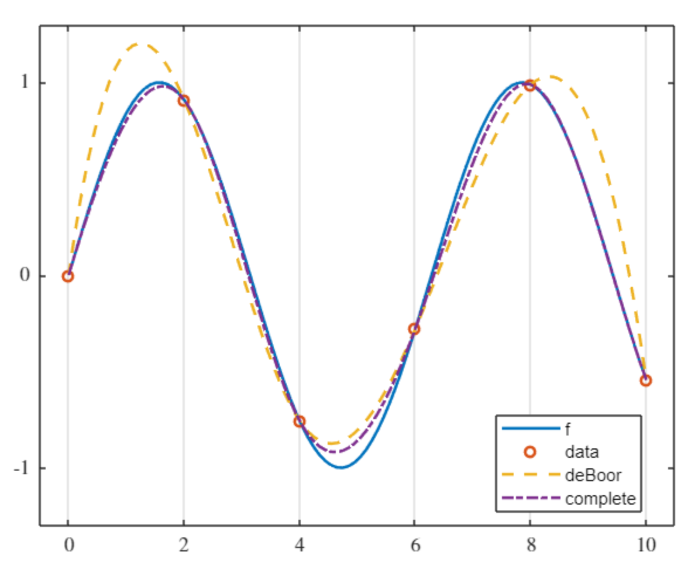
Solution

```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=0:2:10;
fi=f(xi);
x=linspace(0,10,100);
y_f=f(x);
y_deBoor=interp1(xi,fi,x,'spline');
y_complete=spline(xi,[cos(0),fi,cos(10)],x);
plot(x,y_f,'-',xi,fi,'o','MarkerSize',5);
hold on
plot(x,y_deBoor,'--',x,y_complete,'-.')
xticks(xi)
yticks([-1,0,1]);
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-0.5, 10.5, -1.3, 1.3])
legend('f','data','deBoor','complete','Location','best')
hold off

function result = f(x)
result = sin(x);
end
```

“Not-a-knot spline” (Carl de Boor)

Solution



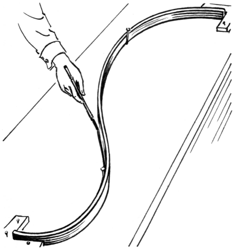
Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

Theorem

For any function $g \in C^2([a,b])$ that interpolates f on Δ, we have

$$\int_a^b (g''(x))^2 dx \geq \int_a^b (s''_{\text{nat}}(x))^2 dx,$$

with equality if and only if $g = s_{\text{nat}}$.



References I

R.T. Trîmbițas
Numerical Analysis in MATLAB
Presa Universitară Clujeană, Cluj-Napoca (2009)

R.L. Burden, D.J. Faires, A.M. Burden
Numerical Analysis
Cengage Learning, Boston (2014)